A TRIDENT SCHOLAR PROJECT REPORT

NO. 476

Solving the Inverse Problem Using Combination Random Graph Models

by

Midshipman 1/C Samuel H. Baker, USN

UNITED STATES NAVAL ACADEMY
ANNAPOLIS, MARYLAND

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)* 5-20-19	**2. REPORT TYPE**	**3. DATES COVERED** *(From - To)*

4. TITLE AND SUBTITLE
Solving the Inverse Problem Using Combination Random Graph Models

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)
Baker, Samuel H.

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
U.S. Naval Academy
Annapolis, MD 21402

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)
Trident Scholar Report no. 476 (2019)

12. DISTRIBUTION / AVAILABILITY STATEMENT

This document has been approved for public release; its distribution is UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

We seek to determine if real-networks can accurately be represented by random graph models. To accomplish this task, we use a combination of three commonly-used random graph models: geometric, Chung-Lu, and preferential attachment. Each of these three models has unique properties that helps model certain characteristics of real-world networks, but using these random graph models individually has proven fruitless. Therefore, we combine multiple models in order to get a model that more accurately reflects these networks. Our method for determining if our combination random graph model successfully represents a real-world network consists of three main tests: edge counts, degree distributions, and triangle counts. This developed algorithm supports the idea that random graph models have potential in modeling real-world networks, and its output is further supported by statistical tests we develop. Although we find some faults in our method, it shows significant potential. We achieved some success with organically produced real-world networks like human and animal social networks and terrorist cells. However, we hypothesize that our model can be improved by adding more random graph models and testing it on larger networks.

15. SUBJECT TERMS
graph theory, network science, social networks, random graphs, Chung-Lu model, geometric model, preferential attachment model

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**		46	**19b. TELEPHONE NUMBER** *(include area code)*

U.S.N.A. --- Trident Scholar project report; no. 476 (2019)

SOLVING THE INVERSE PROBLEM USING COMBINATION RANDOM GRAPH MODELS

by

Midshipman 1/C Samuel H. Baker
United States Naval Academy
Annapolis, Maryland

(signature)

Certification of Adviser Approval

Assistant Professor Franklin H.J. Kenter
Mathematics Department

(signature)

(date)

Acceptance for the Trident Scholar Committee

Professor Maria J. Schroeder
Associate Director of Midshipman Research

(signature)

(date)

Abstract

We seek to determine if real-networks can accurately be represented by random graph models. To accomplish this task, we use a combination of three commonly-used random graph models: geometric, Chung-Lu, and preferential attachment. Each of these three models has unique properties that helps model certain characteristics of real-world networks, but using these random graph models individually has proven fruitless. Therefore, we combine multiple models in order to get a model that more accurately reflects these networks. Our method for determining if our combination random graph model successfully represents a real-world network consists of three main tests: edge counts, degree distributions, and triangle counts. This developed algorithm supports the idea that random graph models have potential in modeling real-world networks, and its output is further supported by statistical tests we develop. Although we find some faults in our method, it shows significant potential. We achieved some success with organically produced real-world networks like human and animal social networks and terrorist cells. However, we hypothesize that our model can be improved by adding more random graph models and testing it on larger networks.

Keywords: graph theory, network science, social networks, random graphs, Chung-Lu model, geometric model, preferential attachment model.

Acknowledgments: We would like to thank the Trident Scholar Program for its support throughout this project. Without the support of the program and committee none of this work would have been possible.

Contents

1 Terminology

- G: fixed simple graph, consisting of vertices and non-directed edges, where an edge is an unordered pair $\{v_i, v_j\}$. We will use the notation used in Chung-Lu, $v_i \sim v_j$, to denote an edge between v_i and v_j [6]. Specifically, we will consider finite, undirected graphs with no loops, unless stated otherwise.

- $V(G)$: vertices of G.

- $E(G)$: Edges of G.

- n: number of vertices in a graph.

- $deg(v)$: The degree of a vertex v in G is defined as the total number of edges incident to v. The total degree of G is defined as the sum of each vertex's individual degree, $\sum_{v \in G} deg(v)$.

- $d(G)$: degree distribution vector of G.

- $\hat{d}(G)$: the sorted degree distribution vector of G. The sorted degree distribution lists the degrees of the vertices of G sorted in descending order. We will let $\hat{d}(G)$ be a vector of length n, where the i-th entry is the i-th highest degree among the vertices of G.

- $e(G)$: edge count of G or the total number of edges in G. The Handshake Lemma states that for undirected simple graphs, the number of degrees is twice the number of edges. Thus,
$$e(G) = \frac{\sum_{v \in G} deg(v)}{2} [16].$$

- *triangle*: a set of three distinct vertices, $\{v_1, v_2, v_3\} \subseteq V(G)$ that are all mutually adjacent to one another.

- $\Delta(G)$: triangle count of G or the total number of unordered triangles found in G. A different ordering of vertices does not constitute a different triangle.

- We approximate several functions in the below sections. Specifically, we say $f(x) \approx g(x)$ if
$$\lim_{x\to\infty} \frac{f(x)}{g(x)} = 1$$
or if,
$$\lim_{x\to 0} \frac{f(x)}{g(x)} = 1.$$

2 Introduction

A “proverb” in network science is that most “real-world” networks are power-law graphs. A graph obeys the power law if the number of vertices with degree k is proportional to $k^{-\beta}$, where $\beta \geq 1$ [6]. Perhaps surprisingly, a recent study of Broido and Clauset found that for a large database of graphs, this proverb, in fact, is not true: only approximately 4% of graphs obey a power-law distribution [5]. There are many processes that generate power-law graphs including preferential attachment [3], hierarchical models [15] and weighted random graphs [6]. One common motif among these processes is that the richer get richer or the popular get more popular. Nonetheless, what if there is some truth behind the idea that real-world networks can be modeled using random graph models? More specifically, it could be the case that the same parameters responsible for generating power-law real-world networks are *partly* responsible for generating other realized networks. For instance, the distribution of words in a language appears to obey the power-law up to a point; after which, other forces appear to take over [4]. We hypothesize that random graph models can in fact represent real-world networks if we combine several models.

In the world of graph theory, there are a lot of ways to define sameness or similarity in graphs. Graphs are often considered the same if they are isomorphic, or there exists some function that can re-order the vertices of one graph so that it is identical to another. However, producing a random graph that is isomorphic to a fixed graph is highly improbable. In fact, even if two random graphs have the same input parameters, achieving an isomorphic relationship is difficult. Nonetheless, two randomly generated graphs produced using the same process typically have similar characteristics (i.e. edge counts and triangle counts) [1]. Therefore, by combining three known random graph models, $G_{\mathbf{w}}$ (or Chung-Lu), geometric, and preferential attachment, we aim to produce a random graph model that can more accurately produce graphs with the same size and similar characteristics of a given network. The challenge here is modeling fixed networks using only three input parameters. Specifically, using three main tests we develop an algorithm to determine if a random graph is similar to a fixed graph, and then test this algorithm on a library of fixed networks. Additionally, we provide a mathematical proof that this algorithm works precisely under mild conditions (Theorems 6.4 and 6.5).

3 Random Graph Models

Formally, we say a random graph model is a random variable that maps from a sample space, S, to a set of possible graphs, $\mathcal{R}$, with specific input parameters, or

$$RGM : S \to \mathcal{R},$$

where RGM is a random graph model. The sample space S is a collection of possible outcomes each with an assigned probability. A random variable is a function mapping from the collection of possible outcomes to another set. In the case of the random graph model, we take the sample space as $[0,1]^k$ for sufficiently large k assigned with usual uniform probability. We then view the random graph model as an interpretation of the point $[0,1]^k$ to turn edges on and off, place vertices, and determine other qualities in the building of the graph. We want to know specific qualities about random graph models, including probability of an edge, edge counts, triangle counts, and degree distributions. These values are random variables as well, mapping from the collection of all possible $R \in \mathcal{R}$ with defined input parameters to the real numbers, or

$$X : \mathcal{R} \to \mathbb{R},$$

where X is the characteristic of R we wish to find. If we combine these two random variables we get

$$X(RGM) : S \to \mathbb{R},$$

where $X(RGM)$ represents a specific characteristic of the random graph model.

This precise definition of a random graph is not needed to understand the remainder of the project, but is necessary for mathematical completeness.

Additionally, we often take the expectation of these characteristics using both mathematical formulas and experimental averages. One important property of expectation that we utilize is linearity of expectation or

$$\mathbb{E}\{aX + bY\} = a\mathbb{E}\{X\} + b\mathbb{E}\{Y\}[7],$$

where $a, b \in \mathbb{R}$ and X and Y are real valued random variables.

3.1 Random Geometric Graphs

The geometric random graph model or $G_{n,r}$ has been studied in depth by Mathew Penrose [14]. The model is simple, but extremely valuable in the study of random graphs because edges are created based on proximity. We build our graph by randomly placing n vertices in the unit square and connect the vertices based on the input parameter r or radius, where $0 \leq r \leq \sqrt{2}$. We form an edge between a pair of vertices in the unit square $\{i, j\}$ if they are less than or equal to a distance r apart.

Theorem 3.1. *For a pair of vertices $\{i, j\} \in V(R)$ generated using the random geometric* *model with parameters n and r,*

$$\pi r^2(1 - 4r + 4r^2) \leq Pr(i \sim j) \leq \pi r^2,$$

where $Pr(i \sim j)$ represents the probability of an edge between $\{i, j\}$. For sufficiently small r, $Pr(i \sim j) \approx \pi r^2$ [14]. That is,

$$\lim_{r \to 0} \frac{Pr(i \sim j)}{\pi r^2} = 1$$

Proof. Place a square with side length $1 - 2r$ within the unit square so that each side of the square is exactly r distance away from the edge of the unit square. Notice that this square has area $(1 - 2r)^2 = (1 - 4r + 4r^2)$. The probability of a vertex falling within this sub-square is equal to the area of the sub-square divided by the area of the unit square. Since the area of the unit square is 1, the probability that a vertex falls within this square is $(1 - 4r + 4r^2)$. The probability that an arbitrary vertex i falls within the sub-square and another arbitrary vertex j falls distance r from vertex i is equal to the probability of i landing in sub-square multiplied by the area of the circle with center i and radius r. Thus,

$$Pr(i \sim j) \geq \pi r^2(1 - 4r + 4r^2)$$

If we assume that an arbitrary vertex i can fall outside of the sub-square and still have probability of πr^2 of another vertex j landing distance r from it, then the upper-bound follows. Since i can land anywhere in the unit square $Pr(i \sim j) \leq \pi r^2$. Thus,

$$\pi r^2(1 - 4r + 4r^2) \leq Pr(i \sim j) \leq \pi r^2.$$

□

Note: the lower bound of this inequality approaches πr^2 as r goes to zero. It is worth mentioning that this the lower bound can be improved by considering specific areas of partial circles near the edge of the square; however, for our application, this precision is not needed. Also, since the r values in the $G_{n,r}$ model are often sufficiently small, it has become common practice in random graph theory to use πr^2 as a sufficient approximation.

Likewise, the expected number of edges can be approximated the same way,

$$\mathbb{E}\{e(R)\} \approx \pi r^2 \binom{n}{2} [14].$$

The proof of this formula can be found in Section 6. The expected number of edges is important for our algorithm described in Section 4. An example of a geometric graph with input parameters $n = 15$ and $r = 0.3$ is pictured below. Note that based on the proof of Theorem 3.1, $r = 0.3$ is not sufficiently small enough to use the approximation $P(i \sim j) = \pi r^2$. However, we use it for ease of presentation in the figure below.

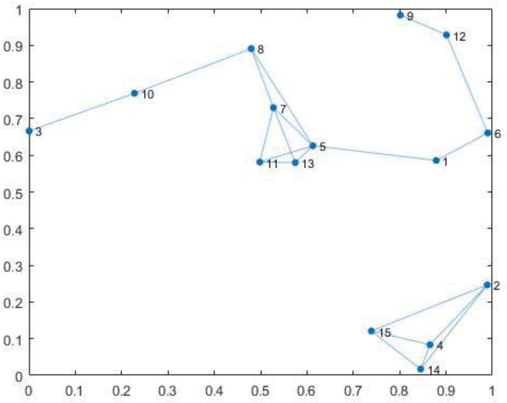


Figure 1: Geometric Random Graph

Because edges are determined by proximity, the likelihood of cliques, complete subgraphs, is high in the geometric model. We can use the geometric model to help simulate a fixed real-world network with several cliques. In our algorithm found in Section 4, we use the number of triangles (three vertex clique) as a method for examining our model.

3.2 $G_{\mathbf{w}}$, Chung-Lu model

The Chung-Lu model first introduced in [6] has a single input parameter: a vector $\mathbf{w}$, where the values of $\mathbf{w}$ are non-negative real numbers. In strict terms, the values in the vector $\mathbf{w}$ are the explicit expected degree of each individual vertex if you allow for self-loops (a vertex can form an edge with itself). When self-loops are allowed, the self-loop only counts as one edge toward the degree of that vertex. For our combination model, we do not wish to use self-loops; however, in most cases, the total number of self-loops is small compared to the total number of edges. Hence in our model, each individual entry of $\mathbf{w}$ is approximately the expected degree contributed by the Chung-Lu model of individual vertices. This result will be proven below.

Given $\mathbf{w}$, where

$$\mathbf{w}_{max}^2 := \left(\max_k \mathbf{w}_k\right)^2 \leq \sum_{k=1}^{n} \mathbf{w}_k, [6]$$

a $G_{\mathbf{w}}$ random graph is generated as follows: For each pair of vertices $\{i, j\}$ place an edge $i \sim j$ with probability

$$Pr(i \sim j) = \frac{\mathbf{w}_i \mathbf{w}_j}{\displaystyle\sum_{k=1}^{n} \mathbf{w}_k}, [6]$$

where each edge is placed independently of all the others. Notice that if $\mathbf{w}$ is a constant vector then $Pr(i \sim j)$ becomes a constant by the above formula. When this occurs, we say the random graph was generated by the $G_{n,p}$ or Erdös-Renyi model, where there are n vertices and a fixed p probability of an edge between any two vertices. We will use the $G_{n,p}$ or Erdős-Renyi model in Section 6.

Theorem 3.2. *For a random graph R generated using the Chung-Lu model with parameter $\mathbf{w}$, the expected edge count obeys the following formula:*

$$\mathbb{E}\{e(R)\} = \frac{\sum_{\substack{i,j=1\\ i\neq j}}^{n} \frac{\mathbf{w}_i \mathbf{w}_j}{\sum_{k=1}^{n} \mathbf{w}_k}}{2} = \frac{\sum_{\substack{i,j=1\\ i\neq j}}^{n} \frac{\mathbf{w}_i \mathbf{w}_j}{\sum_{k=1}^{n} \mathbf{w}_k}}{2\binom{n}{2}} \binom{n}{2}.$$

Additionally,

$$\mathbb{E}\{e(R)\} = \sum_{i=1}^{n} \frac{\mathbf{w}_i}{2} = \sum_{i=1}^{n} \frac{\mathbf{w}_i}{2\binom{n}{2}} \binom{n}{2},$$

if self-loops are permitted.

Proof. Fix i as an arbitrary vertex in R generated by the $G_{\mathbf{w}}$ model with a fixed $\mathbf{w}$.

For each $j \in V(G)$ define,

$$\gamma_{i,j} = \begin{cases} 1 & \text{if edge between } i \text{ and } j \\ 0 & \text{if no edge between } i \text{ and } j \end{cases},$$

where j is an arbitrary vertex in the $G_{\mathbf{w}}$ graph. It follows that

$$\mathbb{E}\{deg(i)\} = \mathbb{E}\{\sum_{\substack{j=1 \\ i \neq j}}^{n} \{\gamma_{i,j}\}\}.$$

By the linearity of expectation,

$$\mathbb{E}\{deg(i)\} = \sum_{\substack{j=1 \\ i \neq j}}^{n} \mathbb{E}\{\gamma_{i,j}\}.$$

By the definition of $Pr(i \sim j)$,

$$\mathbb{E}\{\gamma_{i,j}\} = Pr(i \sim j).$$

Thus,

$$\mathbb{E}\{deg(i)\} = \sum_{\substack{j=1 \\ i \neq j}}^{n} Pr(i \sim j) = \sum_{\substack{j=1 \\ i \neq j}}^{n} \frac{\mathbf{w}_i \mathbf{w}_j}{\sum_{k=1}^{n} \mathbf{w}_k}.$$

Notice,

$$\mathbb{E}\{deg(R)\} = \sum_{i=1}^{n} \mathbb{E}\{\deg(i)\}.$$

Therefore,

$$\mathbb{E}\{deg(R)\} = \sum_{\substack{i=1\\j=1\\i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}.$$

By the Handshake Lemma,

$$\mathbb{E}\{e(R)\} = \frac{\sum_{\substack{i,j=1\\i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}}{2} = \frac{\sum_{\substack{i,j=1\\i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}}{2\binom{n}{2}}\binom{n}{2}.$$

If we allow for self-loops, the condition that $i \neq j$ in the summation goes away. Therefore, factoring out w_i we get,

$$\mathbb{E}\{deg(i)\} = \sum_{j=1}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k}^{n}\mathbf{w}_k} = \mathbf{w}_i\left(\frac{\sum_{j=1}^{n}\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}\right) = \mathbf{w}_i.$$

Therefore,

$$\mathbb{E}\{deg(R)\} = \sum_{i=1}^{n}\mathbf{w}_i.$$

Thus by the Handshake Lemma,

$$\mathbb{E}\{e(R)\} = \sum_{i=1}^{n}\frac{\mathbf{w}_i}{2} = \sum_{i=1}^{n}\frac{\mathbf{w}_i}{2\binom{n}{2}}\binom{n}{2}.$$

□

It is worth remarking that the number of edges between the case of $G_{\mathbf{w}}$ with self-loops and the case without self-loops differs in expectation by

$$\frac{\sum_i \mathbf{w}_i^2}{\sum_k \mathbf{w}_k}.$$

In our model, we use the sorted degree distribution vectors of the fixed network G as our $\mathbf{w}$ for the random graph, R. Even though the $\mathbf{w}$ vector is sorted, each individual entry still represents the approximate expected degree of that vertex as seen by Theorem 3.2. However, we often have to scale down the maximum values of the degree distribution vectors in order to get a $\mathbf{w}$ that meets the necessary condition, $\mathbf{w}_{max}^2 \leq \sum_{k=1}^{n} \mathbf{w}_k$. The main advantage of the Chung-Lu model is that a particular degree sequence can be prescribed *and* independence still holds. The $G_{\mathbf{w}}$ model pictured below has parameters: $n = 15$ and $\mathbf{w} = [5, 5, 5, 4, 4, 4, 3, 3, 3, 2, 2, 2, 1, 1, 1]$.

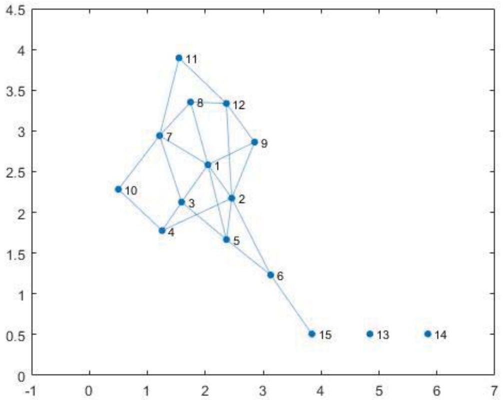


Figure 2: $G_{\mathbf{w}}$ Random Graph

Notice that the expected degree for each vertex, v_i, is approximately its corresponding value $\mathbf{w}_i$ in the vector $\mathbf{w}$. It is important to remember that we are not inputting the exact degree values but rather weights that factor into the probability of an edge.

3.3 Preferential Attachment

The preferential attachment model is an example of an exponential random graph model and is often referred to as the "rich get richer" model. We use Herbert Simon's preferential attachment model of two input parameters, n and k, where n is the number of vertices in the graph and k is the number of edges added at each step [6]. The graph generation process starts with a clique of size k (i.e., k vertices all pair-wise adjacent) and then proceeds to add a vertex. At each step, the added vertex makes k connections to the existing graph based on the existing degrees of the graph. This process is repeated until there are n vertices in the graph. The newly added vertex i will form an edge with an already present vertex, j, with probability

$$Pr(i \sim j) = \frac{deg(j)}{\sum\limits_{j, i \neq j} deg(j)},$$

where $deg(j)$ represents the degree of an existing vertex j in the random graph R. This process is iterative. At each step the $Pr(i \sim j)$ changes because the size and total degree of the graph has changed. If $k > 1$, we use the above formula to assign an edge, but if i forms an edge with j with the first of k edges added at that step, the following k edges cannot duplicate that edge. That is k, distinct edges must form at each step.

Theorem 3.3. *The expected number of edges for the Preferential Attachment model with parameters k and n follows the below formula:*

$$\mathbb{E}\{e(R)\} = k(n-1) = \frac{2k}{n}\binom{n}{2}.$$

Proof. By definition, there are k edges added at each step of the Preferential Attachment process. Since there are $(n-1)$ steps, it follows that

$$\mathbb{E}\{e(R)\} = k(n-1).$$

Therefore,

$$\mathbb{E}\{e(R)\} = \frac{2kn(n-1)}{2n}. = \left(\frac{2k}{n}\right)\left(\frac{n(n-1)}{2}\right) = \left(\frac{2k}{n}\right)\left(\frac{n!}{2!(n-2)!}\right)$$

Thus, by definition of the binomial coefficient,

$$\mathbb{E}\{e(R)\} = k(n-1) = \frac{2k}{n}\binom{n}{2}.$$

□

Note: Our construction of the preferential attachment model occasionally duplicates edges, but this does not effect the results enough to merit any additional proofs.

An example of a preferential attachment graph with parameters $n = 15$ and $k = 1$ is pictured in the figure below.

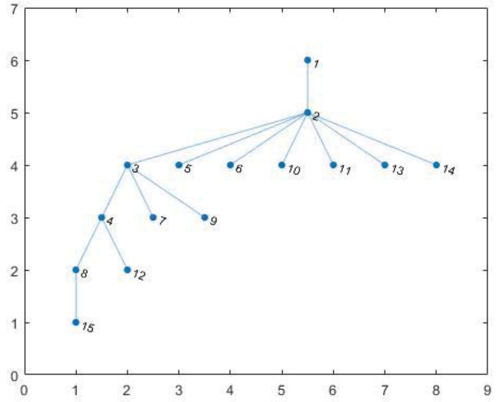


Figure 3: Preferential Attachment Random Graph

If $k = 1$, the model will form a tree as seen above. The advantage of preferential attachment is the development of a central hub of high degree [6].

3.4 Our Combination Model

The geometric, $G_{\mathbf{w}}$, and preferential attachment models all have valuable characteristics that are visible in real world networks, so rather than attempt to model real world networks with a single random graph model, we chose to use a combination of the three random graphs. Essentially, this combination model uses the preferential attachment, geometric and $G_{\mathbf{w}}$ models independently of one another to produce a unique random graph model with parameters r, k, and $\mathbf{w}$. The model begins by creating a preferential attachment graph of n vertices using the inputted k parameter. After that preferential attachment graph is created the vertices are randomly placed in the unit square. The model then adds edges to the graph if either of the conditions for geometric or $G_{\mathbf{w}}$ are met. That is if a vertex j falls within distance r of vertex i or based on the $Pr(i \sim j)$ formula for $G_{\mathbf{w}}$ with an inputted

$\mathbf{w}$. For emphasis, to maintain independence, the indices of $\mathbf{w}$ compared to the indices of the preferential attachment process are random. It is important to note that if both conditions are met, then only one edge is formed between i and j. Therefore, edges are drawn using all three input parameters, k, r, and $\mathbf{w}$, but only one needs to be satisfied for the edge to exist.

Theorem 3.4. *The expected number of edges for our combination model with parameters n, k, and $\mathbf{w}$, all fixed, and sufficiently small r, has the following approximation:*

$$\begin{aligned}\mathbb{E}\{e(R)\} \quad\approx\quad & \Bigg(\pi r^2 + \frac{2k}{n} + \frac{\sum_{\substack{i,j=1\\ i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}}{2\binom{n}{2}} - \pi r^2 \frac{\sum_{\substack{i,j=1\\ i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}}{2\binom{n}{2}} \\ & - \frac{2k}{n} \frac{\sum_{\substack{i,j=1\\ i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}}{2\binom{n}{2}} - \frac{2k}{n}\pi r^2 + \frac{2k}{n}\pi r^2 \frac{\sum_{\substack{i,j=1\\ i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}}{2\binom{n}{2}}\Bigg)\binom{n}{2}.\end{aligned}$$

Proof. Notice that the expected edge counts for each individual model are of the form:

$$p\binom{n}{2},$$

where p is the probability or the average probability of an edge.

Note: the average probability refers to the preferential attachment model. Because the probability of an edge changes at each step we average the probabilities as $\frac{2k}{n}$.

Therefore, we need to determine the probability of an edge in our combination model. The probability of an edge for our model must be a linear combination of the three individual probabilities. By the inclusion-exclusion principle we know,

$$\begin{aligned}(Pr_{Geo} \cup Pr_{PA} \cup Pr_{G_{\mathbf{w}}}) = Pr_{Geo} + Pr_{PA} + Pr_{G_{\mathbf{w}}} - (Pr_{Geo} \cap Pr_{G_{\mathbf{w}}}) \\ -(Pr_{PA} \cap Pr_{G_{\mathbf{w}}}) - (Pr_{Geo} \cap Pr_{PA}) + (Pr_{Geo} \cap Pr_{PA} \cap Pr_{G_{\mathbf{w}}}),\end{aligned}$$

where each of the above probabilities represent the probability of an edge for that particular model. Since the models determine edges independently of one another the formula simplifies,

$$Pr_{Total} = Pr_{Geo}+Pr_{PA}+Pr_{G_{\mathbf{w}}}-(Pr_{Geo}Pr_{G_{\mathbf{w}}})-(Pr_{PA}Pr_{G_{\mathbf{w}}})-(Pr_{Geo}Pr_{PA})+(Pr_{Geo}Pr_{PA}Pr_{G_{\mathbf{w}}}),$$

where $Pr_{Total} = (Pr_{Geo} \cup Pr_{PA} \cup Pr_{G_{\mathbf{w}}})$. Therefore,

$$\begin{aligned} Pr_{total} \;\approx\; & \Bigg(\pi r^2 + \frac{2k}{n} + \frac{\sum_{\substack{i,j=1\\ i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}}{2\binom{n}{2}} - \pi r^2 \frac{\sum_{\substack{i,j=1\\ i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}}{2\binom{n}{2}} \\ & -\frac{2k}{n}\frac{\sum_{\substack{i,j=1\\ i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}}{2\binom{n}{2}} - \frac{2k}{n}\pi r^2 + \frac{2k}{n}\pi r^2 \frac{\sum_{\substack{i,j=1\\ i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}}{2\binom{n}{2}}\Bigg), \end{aligned}$$

for sufficiently small r. Thus,

$$\begin{aligned} \mathbb{E}\{e(R)\} \;\approx\; & \Bigg(\pi r^2 + \frac{2k}{n} + \frac{\sum_{\substack{i,j=1\\ i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}}{2\binom{n}{2}} - \pi r^2 \frac{\sum_{\substack{i,j=1\\ i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}}{2\binom{n}{2}} \\ & -\frac{2k}{n}\frac{\sum_{\substack{i,j=1\\ i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}}{2\binom{n}{2}} - \frac{2k}{n}\pi r^2 + \frac{2k}{n}\pi r^2 \frac{\sum_{\substack{i,j=1\\ i\neq j}}^{n} \frac{\mathbf{w}_i\mathbf{w}_j}{\sum_{k=1}^{n}\mathbf{w}_k}}{2\binom{n}{2}}\Bigg)\binom{n}{2}. \end{aligned}$$

for sufficiently small r. □

The reason the formula is an approximation is simply because we are using some of the $\mathbb{E}\{e(R)\}$ approximations from the individual models. We have found that this formula is more accurate for random graphs with large n and small r. An example of this model with input parameters $n = 15$, $k = 1$, $r = 0.2$, and $\mathbf{w} = [5, 5, 5, 4, 4, 4, 3, 3, 3, 2, 2, 2, 1, 1, 1]$ is pictured below.

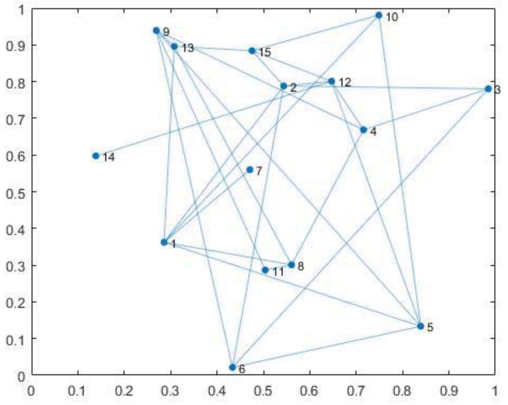


Figure 4: Combination Model Random Graph

Using this new random graph model, we believe it is possible to accurately model a multitude of real-world networks.

4 Solving the Inverse Problem

4.1 Problem Statement

Given a fixed graph G, we seek to find the most ideal k, $\mathbf{w}$, and r in our combination model that are best able to produce random graphs, R, like G. However, in order to simplify the problem, we limited the broadness of these three parameters. We are testing only values of $k \in \{0, 1, 2\}$. Additionally, we define $\mathbf{w}$ to be a fixed multiple of the degree distribution vector of G. We define $\alpha \in [0, 1]$ as the coefficient of $\mathbf{w}$. It might seem like we are using the answer to find the answer. However, our studies have shown that not all distribution vectors meet the requirements of $\mathbf{w}$ set out in Chung-Lu, so we have to modify the vector to meet those criteria. Finally, we limit $r \in [0, 1]$.

Therefore, a more accurate problem statement: Given a fixed graph G, we seek to find the most ideal $k \in \{0, 1, 2\}$, $\alpha \in [0, 1]$, and $r \in [0, 1]$ in our model that are best able to produce random graphs, R, like G. The difficulty in this objective is determining how to measure similarity. Because our model is by definition random, we face the impractical task of comparing a set of random graphs to a fixed network. However, despite the randomness of the individual graphs, when the whole of these individual graphs is examined we find common characteristics. Consequently, our goal is not to find the graph that most closely represents the fixed graph G. If that were the case, we would simply choose a unique fixed graph to model G. Instead, our goal is to determine if the underlying structure of real world networks can be described using random graph models. In order to do this we must determine how we will measure success, which will be discussed in the following subsection.

4.1.1 Overview

Our process for determining whether random graphs can successfully represent real-world networks begins with determining our measures of success. We then vary the input parameters of our combination model until we find a set of input parameters that best represents the fixed graph based on the measures of success.

4.1.2 Measures

Before we begin in detailing our process for solving the problem statement, we must define the measures used to determine success. Comparing a fixed graph to a random graph and determining similarity is a difficult task, and the ultimate measure of closeness is not agreed upon. Therefore, we decided to use three main measures of success.

- $Edge_{\%}$:

  $$Edge_{\%} = \frac{|\mathbb{E}\{e(R)\} - e(G)|}{e(G)}.$$

  This formula represents the relative error between the expected edge count of an individual random graph, R and the actual number of edges in the fixed graph G.

- $Trig_{\%}$:

  $$Trig_{\%} = \frac{|\Delta(R) - \Delta(G)|}{\Delta(G)}.$$

Similar to the formula for $Edge_{\%}$, this formula represents the relative error between the triangle count of an individual random graph, R and the actual number of triangles in the fixed graph G. Among small random graphs the triangle counts vary dramatically, so this value is often high.

- $Deg_{\%}$:

$$Deg_{\%} = \frac{\sqrt{\sum_j^n (\hat{d}_j(G) - \hat{d}_j(R))^2}}{n}.$$

  $Deg_{\%}$ is a semi-norm, defined as a norm without the $\|x\| = 0$ if and only if $x = 0$ condition, comparing the degree distribution vectors of G and R. The numerator of this fraction compares the sorted degree distribution vector of G with the sorted degree distribution vector of an individual random graph using the common 2-norm. We found that if the vectors are compared using only the 2-norm, the norm values vary too much for meaningful results. By dividing by the number of vertices, we achieve a more consistent comparison over graphs of different sizes.

- Note: The expectation in $Edge_{\%}$ is calculated using the formula in Section 3. $Trig_{\%}$ and $Deg_{\%}$ are both random variables, and therefore the values change for every random graph R. In order to get more precise results, we approximate $Trig_{\%}$ and $Deg_{\%}$ via a simulation of 100 trials.

4.1.3 Algorithm: Inputs

- G: the fixed real-world network

- ε: the precision threshold for α and r; to make the algorithm more efficient, we start testing only the tenth values from 0 to 1 for α and r. In an iterative process, we then narrow down this interval. ε is the minimum we wish to narrow the interval to (i.e. for an r or α value to the nearest thousandth, $\varepsilon = 0.001$).

- $MaxEdge_{\%}$: the maximum relative error between the edge counts of R and G to be considered successful

- $MaxTrig_{\%}$: the maximum relative error between the number of triangles of R and G to be considered successful.

- $MaxDeg_{\%}$: the maximum semi-norm value allowed between the degree distribution vectors of R and G to be considered successful.

- Note: Determining what is a reasonable value for $Edge_{\%}$, $Trig_{\%}$, and $Deg_{\%}$ is a topic discussed more in section 6.2.

4.1.4 Algorithm: Process

The overall idea of this algorithm is to take a systematically chosen collection of the possible combinations of k, α, and r and test them for success based off our defined criteria. Throughout the process we narrow the number of parameters we are testing by discarding the unsuccessful combinations of k, α, and r, keeping only those parameters that meet our criteria.

Using the formula for the expected number of edges of our combination model we generate three arrays. Each one of the entries in the three arrays is the expected number of edges for our combination model given these k, α and r values. Notice that the precision interval of α and r starts at 0.1, and the three arrays each have a different k value based off the three possible k values.

$$\begin{bmatrix} (k=0,\alpha=0,r=0) & (k=0,\alpha=0.1,r=0) & \dots & (k=0,\alpha=1,r=0) \\ (k=0,\alpha=0,r=0.1) & (k=0,\alpha=0.1,r=0.1) & \dots & (k=0,\alpha=0,r=0.1) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (k=0,\alpha=0,r=1) & (k=0,\alpha=0.1,r=1) & \dots & (k=0,\alpha=1,r=1) \end{bmatrix}$$

$$\begin{bmatrix} (k=1,\alpha=0,r=0) & (k=1,\alpha=0.1,r=0) & \dots & (k=1,\alpha=1,r=0) & \\ (k=1,\alpha=0,r=0.1) & (k=1,\alpha=0.1,r=0.1) & \dots & (k=1,\alpha=0,r=0.1) & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (k=1,\alpha=0,r=1) & (k=1,\alpha=0.1,r=1) & \dots & (k=1,\alpha=1,r=1) & \end{bmatrix}$$

$$\begin{bmatrix} (k=2,\alpha=0,r=0) & (k=2,\alpha=0.1,r=0) & \dots & (k=2,\alpha=1,r=0) & \\ (k=2,\alpha=0,r=0.1) & (k=2\alpha=0.1,r=0.1) & \dots & (k=2,\alpha=0,r=0.1) & \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (k=2,\alpha=0,r=1) & (k=2,\alpha=0.1,r=1) & \dots & (k=2,\alpha=1,r=1) & \end{bmatrix}$$

The expected edge count for each random graph produced with the parameters in an entry is then compared to $e(G)$. If $Edge_{\%} \leq MaxEdge_{\%}$, then that set of parameters is saved for the next step. If this condition is not met, then that set of parameters is dismissed. Once each individual entry of all three matrices is tested, we decrease the precision interval based off the inputted ε. We will call this intermediate precision interval ε_r for the r values and ε_α for the α values. We now sort through every set of k, r, and α values that met $Edge_{\%} \leq MaxEdge_{\%}$ in order to find the maximum and minimum r and α values. Using ε_r, we divide the interval between $r_{min} - \varepsilon_r$ and $r_{max} + \varepsilon_r$ in to ten equally spaced values. The same is done for the α values using ε_α. The new ε_r and ε_α become the interval between these new r and α values. This process is continued as long as $\varepsilon \leq \varepsilon_r$ and $\varepsilon \leq \varepsilon_\alpha$. For example, let's say we are only concerned with r and not k and α. If $r = 0.1$ and $r = 0.4$ both meet $Edge_{\%} \leq MaxEdge_{\%}$ when $\varepsilon_r = 0.1$, then we would divide the interval from 0 to 0.5 into 10 equal spaced values and the new $\varepsilon_r = 0.0556$. There are specific steps followed in unique cases such as when r_{max} or α_{max} equal 1 and r_{min} or α_{min} equal 0, but overall the process runs as stated above.

The next step begins by taking the first saved set of k, α, and r of the finest interval from the previous step. Using these parameters we generate a random graph using our combination random graph model. We then calculate $\Delta(R)$ and compare it to $\Delta(G)$ using the $Trig_{\%}$ measure. Similarly, we calculate $Trig_{\%}$ using the associated semi-norm. If both $Deg_{\%} \leq MaxDeg_{\%}$ AND $Trig_{\%} \leq Max\%$ are met then that individual trial is deemed a success and we initialize a counter, $C = 1$. We repeat this process for this same set of input parameters 100 times, and for each successful trial increment C by 1. We choose 100 trials for ease of computation. We then divide the total number of successful trials by the total number of trials to get the fraction of successful trials. We call this value N. This gives us a number between 0 and 1 that measures how successful that individual set of input parameters was at modeling the fixed network G using the defined measures. We repeat this process for all of the save input parameters, thus getting an N value for each. We know that N is a random variable; however, we have found through experimentation that they have a low variance. A specific example is provided in Section 5.

4.1.5 Algorithm: Outputs

We output the α, r, and k that achieved the highest N value. An N value close to one means that there exists a set of input parameters for our combination model that is relatively similar to the fixed network G using our defined measures.

A schematic of the whole process can be seen below.

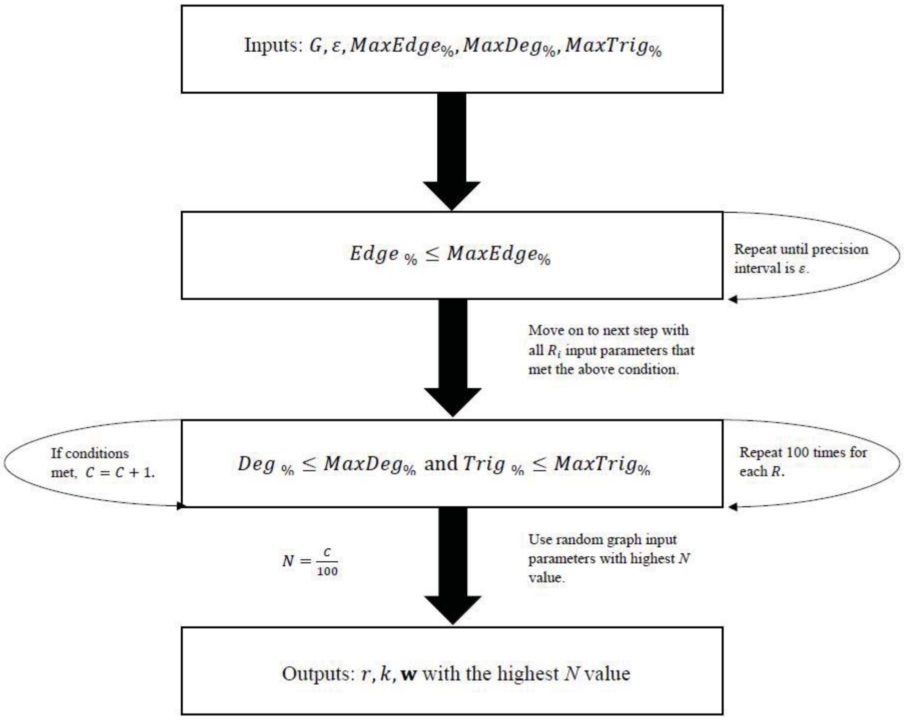

Inputs: $G, \varepsilon, MaxEdge_{\%}, MaxDeg_{\%}, MaxTrig_{\%}$
$Edge_{\%} \leq MaxEdge_{\%}$
Repeat until precision interval is ε.
Move on to next step with all R_i input parameters that met the above condition.
If conditions met, $C = C + 1$.
$Deg_{\%} \leq MaxDeg_{\%}$ and $Trig_{\%} \leq MaxTrig_{\%}$
Repeat 100 times for each R.
$N = \frac{C}{100}$
Use random graph input parameters with highest N value.
Outputs: $r, k, \mathbf{w}$ with the highest N value

Figure 5: Our Algorithm

5 Results

5.1 In-Depth: Zachary Karate Club

The Zachary Karate Club graph is a 34-vertex graph that represents the separation of two karate clubs [18]. In this graph a vertex represents a person and an edge represents two people knowing each other. Although this graph is small, it provides us with a way to test our method quickly on a basic social network. Through experimentation, we have found that the best results are achieved when,

- $\varepsilon = 0.01$: Smaller division intervals do not yield significantly better results.
- $MaxEdge_{\%} = 0.08$: Decreasing the edge percent will decrease the number of combinations sent to the next two steps, but experimentation shows that better results can be achieved if we keep the edge percent a little bit larger.

- $MaxDeg_{\%} = 0.29$: 0.29 might seem too high to indicate any type of significant result, but the larger $Deg_{\%}$ value is due to the small size of the graph. As the number of vertices gets larger, we can decrease this value.

- $MaxTrig_{\%} = 0.39$: Triangle counts vary significantly in small random graphs, so we had to set the $MaxTrig_{\%}$ higher to allow for this variance.

Note: We understand that these values seem arbitrary. We could easily set each measure high and achieve "successful" results. However, we found through experimentation over several different input parameters that these input parameters achieve the best results. That is they achieve the random graph that most closely relates to the fixed real-world network. Further justification can be found at the end of Section 6.

After running our algorithm with these inputs, we maintained consistent outputs over several trials. The ideal R found had these parameters:

- $N = 0.75$ with $\sigma^2 = 0.0012$

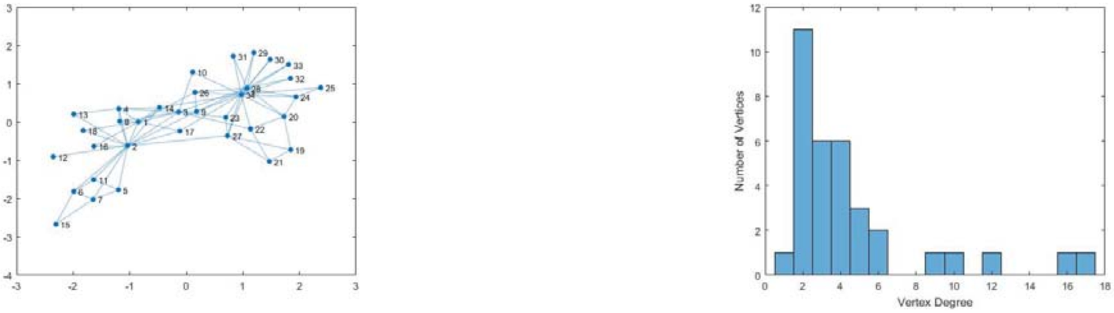
- $r = 0.0104$ with $\sigma^2 = 1.1489e - 4$

- $k = 1$

- $\alpha = 0.7496$ with $\sigma^2 = 5.9009e - 5$

Note: These parameters are the average values produced after running the algorithm 100 times for the network, and σ^2 represents the variance of the values over the simulation.

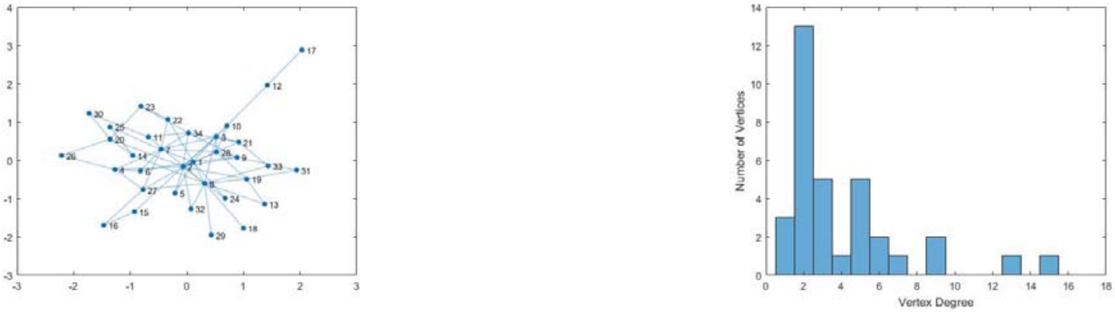
Zachary's Karate network is a small network, so the fact that we were able to model it so closely with random graphs validates the idea behind our algorithm. Our method was able to produce a set of random graph parameters that meet the specified measures 75% of the time. Additionally because our process uses random graphs to model fixed networks, we now have multiple graphs that can model Zachary's Karate network. The next several figures demonstrate this fact.



(a) Zachary's Karate Club Graph

(b) Zachary's Karate Club Histogram

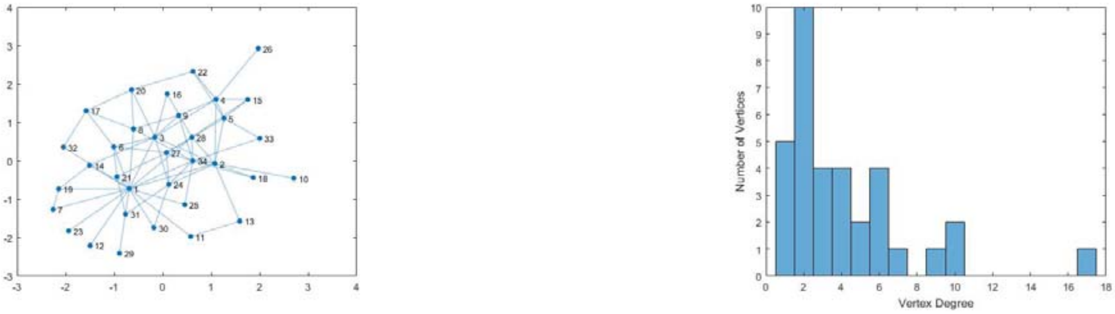
Figure 6: Zachary's Karate Club



(a) R Graph 1

(b) R Histogram 1

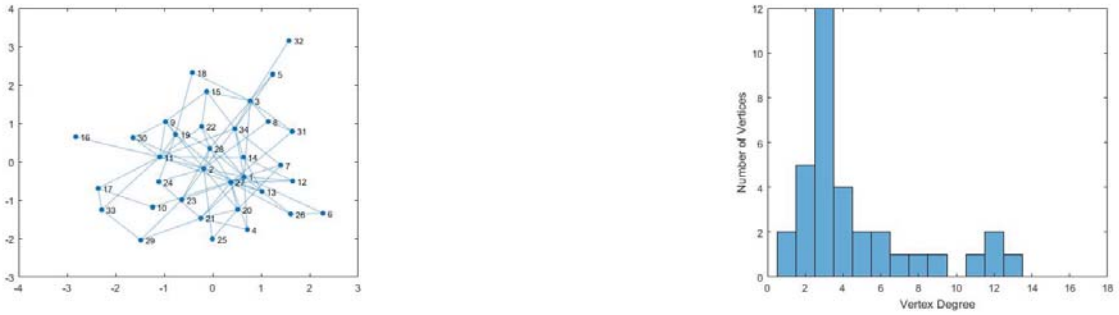
Figure 7: Random Graph 1 Representation of Karate Club Network



(a) R Graph 2

(b) R Histogram 2

Figure 8: Random Graph 2 Representation of Karate Club Network



(a) R Graph 3

(b) R Histogram 3

Figure 9: Random Graph 3 Representation of Karate Club Network

5.2 Comparison To A Library Of Graphs

Despite positive results with Zachary's Karate Club network, our algorithm is worth very little if it cannot produce positive results over a larger library of graphs. In order to further our study, we ran our algorithm on an additional seven fixed networks varying in size from 62 to 126 vertices. The results can be found in the table below.

Network	**Size**	$MaxDeg_{\%}$	$MaxTrig_{\%}$	N	k	r	α	**Source**
Dolphins	$n = 62$	0.26	0.35	0.76	1	0.1366	0.0878	[13]
France High School	$n = 126$	0.28	0.36	0.93	1	0.0329	1.0000	[8]
Hamburg Terrorist	$n = 77$	0.30	0.90	0.26	2	0.0165	0.5463	[2]
Les Miserables	$n = 77$	0.24	0.62	0.77	0	0.0658	1.0000	[11]
Madrid Train	$n = 64$	0.21	0.62	0.74	0	0.0782	0.9753	[10]
911 Terrorist	$n = 62$	0.25	0.57	0.78	0	0.0988	0.7942	[12]
Star Wars	$n = 110$	0.23	0.50	0.73	1	0.0165	0.9012	[9]

Table 1: Results From Our Library Of Tested Graphs

We understand that some of the $MaxDeg_{\%}$ values and $MaxTrig_{\%}$ values seem very high. However, we justify these inputs in Section 6. The table provides valuable results because it demonstrates that our process is successful over several graphs. With the exception of the Hamburg Terrorist network, all of the graphs had relatively high N values.

6 Proof of Concept

We have shown through experimentation that our algorithm is capable of producing random graphs that meet our measures of success, but it is important now to justify why we chose these measures and their associated max values. In this section we will show that under mild assumptions, the geometric model will almost certainly have more triangles than the $G_{\mathbf{w}}$ model for a constant $\mathbf{w}$. In other words, given a typical graph determined by the geometric model the algorithm from Section 4 will almost certainly distinguish it from a random graph determined by the $G_{\mathbf{w}}$ model with constant $\mathbf{w}$.

6.1 Triangle Count

We propose that the triangle count is a necessary test for determining the best parameters for our model because it helps distinguish between the random edges found in the $G_{\mathbf{w}}$ model and the edges determined by proximity found in the geometric model. We suggest that the geometric model will produce more triangles than the $G_{\mathbf{w}}$ model. This is a very significant hypothesis, but can be proven rather quickly under certain assumptions. First, we assume that the $G_{\mathbf{w}}$ model has a constant $\mathbf{w}$ vector, which is the traditional $G_{n,p}$ or Erdös-Renyi model. We must first prove the expected edge count and expected triangle counts for each of these models before we can prove that the geometric model will produce more triangles than the $G_{n,p}$ model.

Theorem 6.1. *For fixed p, the expected number of edges for a $G_{n,p}$ is as follows:*

$$\mathbb{E}\{e(R)\} = p\binom{n}{2},$$

where p is the inputted probability of an edge between two vertices. For sufficiently large n, *the expected edge count can be approximated to be $\frac{pn^2}{2}$. That is,*

$$\lim_{n\to\infty} \frac{\mathbb{E}\{e(R)\}}{\frac{pn^2}{2}} = 1.$$

Proof. Given two vertices, u and v, in the $G_{n,p}$ graph define,

$$\Omega_{u,v} = \begin{cases} 1 & \text{if edge between } u \text{ and } v \\ \\ 0 & \text{if no edge between } u \text{ and } v \end{cases},$$

Then, it follows that

$$e(R_i) = \sum_{\{u,v\}} \Omega_{u,v}$$

where $e(R_i)$ is the is the number of edges for each random graph and $\{u, v\}$ are all not ordered pairs of u and v. Therefore,

$$\mathbb{E}\{e(R)\} = \mathbb{E}\Big[\sum_{\{u,v\}} \Omega_{u,v}\Big].$$

By the linearity of expectation,

$$\mathbb{E}\{e(R)\} = \sum_{\{u,v\}} \mathbb{E}[\Omega_{u,v}].$$

By definition of p,

$$\mathbb{E}\{e(R)\} = \sum_{\{u,v\}} p.$$

Since there are $\binom{n}{2}$ pairs of edges,

$$\mathbb{E}\{e(R)\} = p\binom{n}{2}.$$

With sufficiently large n,

$$\mathbb{E}\{e(R)\} = p\binom{n}{2} = \frac{pn(n-1)}{2} \approx \frac{pn^2}{2}.$$

□

Theorem 6.2. *For sufficiently small r, the expected number of edges for a random graph produced using the $G_{n,r}$ model is as follows:*

$$\mathbb{E}\{e(R)\} \approx \pi r^2 \binom{n}{2}.$$

With sufficiently large n, the expected edge count can be approximated further as $\frac{\pi r^2 n^2}{2}$.

Proof. In Section 3 we proved that

$$P(u \sim v) \approx \pi r^2.$$

With the probability of an edge determined, the proof follows the same argument as the proof for Theorem 6.1 by replacing p with πr^2. Thus,

$$\mathbb{E}\{e(R)\} \approx \pi r^2 \binom{n}{2}$$

With sufficiently large n, and sufficiently small r,

$$\mathbb{E}\{e(R)\} \approx \pi r^2 \binom{n}{2} = \frac{\pi r^2 n(n-1)}{2} \approx \frac{\pi r^2 n^2}{2}.$$

□

Theorem 6.3. *With fixed p, the expected of triangles in the $G_{n,p}$ is as follows:*

$$\mathbb{E}\{\Delta(R)\} = p^3 \binom{n}{3}.$$

With sufficiently large n this value can be approximated to $\frac{p^3n^3}{6}$.

Proof. Define,

$$\Psi_{u,v,w} = \begin{cases} 1 & \text{if triangle between } u,\ v, \text{ and } w \\ \\ 0 & \text{if no triangle between } u,\ v, \text{ and } w \end{cases},$$

where u, v, and w are arbitrary vertices in the $G_{n,p}$ graph. Then, it follows that

$$\Delta(R) = \sum_{\{u,v,w\}} \Psi_{u,v,w}.$$

Where $\Delta(R)$ is the number of triangles in a random graph, and $\{u, v, w\}$ is the number of unordered trios of u, v, and w. Therefore,

$$\mathbb{E}\{\Delta(R)\} = \mathbb{E}\Big[\sum_{\{u,v,w\}} \Psi_{u,v,w}\Big].$$

By the linearity of expectation,

$$\mathbb{E}\{\Delta(R)\} = \sum_{\{u,v,w\}} \mathbb{E}[\Psi_{u,v,w}].$$

For the three vertices, u, v, w, we know there are $\binom{3}{2}$ or 3 independent pairs of vertices. By definition, the probability that an edge forms between any of those pairs is p. Therefore, the probability that a triangle forms between u, v, and w is p^3. Hence,

$$\mathbb{E}\{\Delta(R)\} = \sum_{\{u,v,w\}} \mathbb{E}[\Psi_{u,v,w}] = \sum_{trios(u,v,w)} p^3.$$

Since there are $\binom{n}{3}$ possible trios of vertices in the $G_{n,p}$ graph,

$$\mathbb{E}\{\Delta(R)\} = p^3\binom{n}{3}.$$

Thus with sufficiently large n,

$$\mathbb{E}\{\Delta(R)\} = p^3\binom{n}{3} = \frac{p^3 n(n-1)(n-2)}{6} = \frac{p^3(n^3 - 3n + 2n)}{6} \approx \frac{p^3 n^3}{6}.$$

□

Theorem 6.4. *For fixed $r \leq \frac{1}{2}$ and sufficiently large n in the $G_{n,r}$ model $\mathbb{E}\{\Delta(R)\}$ obeys*

$$\left(\frac{\pi r^4 n^3}{6}\right)\gamma \leq \mathbb{E}\{\Delta(R)\} \leq \left(\frac{\pi r^4 n^3}{6}\right)\pi,$$

where $\gamma = \frac{2\pi}{3} - \frac{\sqrt{3}}{2} \approx 1.22837$.

Proof. We will begin by proving the lower bound. Under the assumption of the unit square, we know that given a random vertex, u in the $G_{n,r}$ model the probability of another vertex, v landing within the radius, r, of u is πr^2. Note that if a third vertex, w falls within both the radius of u and v, then a triangle will form. Therefore, to find a maximum lower bound we have to determine the location of v within the radius of u that results in the least amount of overlap between the two circles with equal radius. This occurs when v lands exactly distance r away from u. See the below figure.

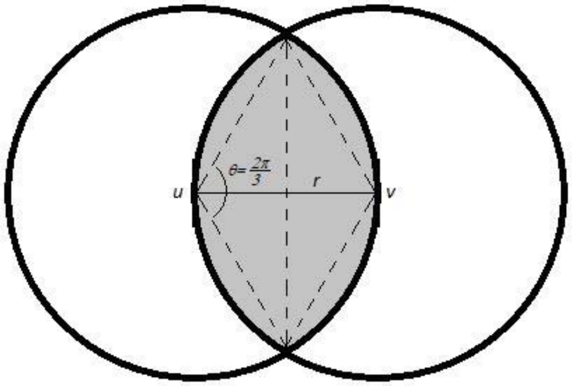


Figure 10: Geometric Lower Bound Triangle Count Area Representation

Based off the formula for the area of a circle segment,

$$A = \frac{r^2}{2}(\theta - \sin\theta), [17]$$

We get that the area for the overlap of the two circles is $2(\frac{r^2\gamma}{2}) = r^2\gamma$, where $\gamma = \frac{2\pi}{3} - \frac{\sqrt{3}}{2}$. We know already that the probability of an edge forming between two vertices in a geometric graph is πr^2. This is due to the fact that the area of the unit square is 1, and therefore, the probability that a point falls within a certain area is just that area. Hence, the probability that w lands within the shaded area in the above figure is $r^2\gamma$. Thus, the probability of all three vertices forming a triangle is $(r^2\gamma)(\pi r^2) = r^4\pi\gamma$. If we replace the probability of a triangle for $G_{n,p}$, p^3, found in the proof of Theorem 6.3 with $r^4\pi\gamma$, the lower bound for the expected number of triangles in the geometric model is

$$\pi r^4\gamma\binom{n}{3} = \frac{\pi r^4\gamma n(n-1)(n-2)}{6} = \frac{\pi r^4\gamma(n^3 - 3n + 2n)}{6} \approx \Big(\frac{\pi r^4 n^3}{6}\Big)\gamma.$$

for sufficiently large n.

Similarly, the greatest possible overlap between the circles of u and v occurs when they land exactly on top of one another. If this occurs then the area w must fall within is simply πr^2. Thus, the expected number of triangles becomes

$$(\pi r^2)^2\binom{n}{3} = \frac{\pi^2 r^4 n(n-1)(n-2)}{6} \approx \Big(\frac{\pi r^4 n^3}{6}\Big)\pi,$$

with sufficiently large n.

Therefore,

$$\frac{\pi r^4\gamma n^3}{6} \leq \mathbb{E}\{\Delta(R)\} \leq \frac{\pi^2 r^4 n^3}{6},$$

where $\gamma = \frac{2\pi}{3} - \frac{\sqrt{3}}{2}$. □

To prove that for large n the geometric model has more triangles than the $G_{n,p}$ model is difficult without certain assumptions. Mainly, since the focus of our test is to distinguish how much each model contributes to the make up of the desired graph, we have to compare the triangle counts of geometric and $G_{n,p}$ with equal $\mathbb{E}\{e(R)\}$. This can be achieved by taking $p = \pi r^2$. Additionally, because the expected number of triangles for the geometric model is a range, we can only prove that the Geometric model has more triangles than the $G_{n,p}$ model for particular r values.

Theorem 6.5. *Given $p = \pi r^2$ and a fixed $r \leq \frac{\sqrt{\frac{2\pi}{3} - \frac{\sqrt{3}}{2}}}{\pi}$, the geometric model will asymptotically almost surely produce more triangles than the $G_{n,p}$ model for a sufficiently large n.*

Proof. In order to prove this theorem, we must first prove that the expected number of triangles for $G_{n,p}$ is less than the lower bound of the expected number of triangles for the geometric model. Then we must prove that this holds asymptotically almost surely. Assume,

$$r \leq \sqrt{\frac{\frac{2\pi}{3} - \frac{\sqrt{3}}{2}}{\pi^2}} = \frac{\sqrt{\frac{2\pi}{3} - \frac{\sqrt{3}}{2}}}{\pi}.$$

Then,

$$\pi^2 r^2 \leq \frac{2\pi}{3} - \frac{\sqrt{3}}{2}.$$

Since $\gamma = \frac{2\pi}{3} - \frac{\sqrt{3}}{2}$,

$$\pi^2 r^2 \leq \gamma.$$

Taking r to be fixed, it follows that

$$\pi^2 r^2 \frac{\pi r^4 n^3}{6} \leq \gamma \frac{\pi r^4 n^3}{6}.$$

Rearranging we get,

$$\frac{(\pi r^2)^3 n^3}{6} \leq \frac{\pi r^4 \gamma n^3}{6}.$$

Therefore by applying Theorems 6.3 and 6.4,

$$\mathbb{E}\{\Delta(R)_{G_{n,p}}\} = p^3\binom{n}{3} \approx \frac{p^3n^3}{6} = \frac{(\pi r^2)^3 n^3}{6} \leq (r^2\gamma)(\pi r^2)\binom{n}{3} =$$

$$\frac{\pi r^4 \gamma n(n-1)(n-2)}{6} \approx \frac{\pi r^4\gamma n^3}{6} \leq \mathbb{E}\{\Delta(R)_{G_{n,r}}\},$$

where $\gamma = \frac{2\pi}{3} - \frac{\sqrt{3}}{2}$.

Thus, the Geometric model will produce more triangles than the $G_{n,p}$ model while,

$$r \leq \sqrt{\frac{\frac{2\pi}{3} - \frac{\sqrt{3}}{2}}{\pi^2}} = \frac{\sqrt{\frac{2\pi}{3} - \frac{\sqrt{3}}{2}}}{\pi} \approx 0.352789.$$

In order to show that this asymptotically almost surely holds, Alon-Spencer tell us it is sufficient to prove that [1]

$$\frac{\mathrm{Var}\{\Delta(R)\}}{\mathbb{E}\{\Delta(R)\}^2} \to 0 \text{ as } n \to \infty$$

for both the $G_{n,p}$ and geometric models. By definition of variance,

$$\mathrm{Var}\{\Delta(R)\} = \mathbb{E}\{\Delta(R)^2\} - \mathbb{E}\{\Delta(R)\}^2.$$

Define,

$$\Psi_{u,v,w} = \begin{cases} 1 & \text{if triangle between } u, v, \text{ and } w \\ 0 & \text{if no triangle between } u, v, \text{ and } w \end{cases},$$

where u, v, and w are arbitrary vertices in the random graph R. Let S and T be two arbitrary unordered trios of vertices in the random graph R. Then,

$$\mathrm{Var}\{\Delta(R)\} = \mathbb{E}\left(\sum_S \Psi_S \sum_T \Psi_T\right) - \mathbb{E}\left(\sum_S \Psi_S\right)\mathbb{E}\left(\sum_T \Psi_T\right).$$

By combing the first two sums,

$$\mathrm{Var}\{\Delta(R)\} = \mathbb{E}\left(\sum_{S,T} \Psi_S \Psi_T\right) - \mathbb{E}\left(\sum_{S} \Psi_S\right) \mathbb{E}\left(\sum_{T} \Psi_T\right).$$

By linearity of expectation,

$$\mathrm{Var}\{\Delta(R)\} = \sum_{S,T} \mathbb{E}\{\Psi_S \Psi_T\} - \sum_{S} \mathbb{E}\{\Psi_S\} \sum_{T} \mathbb{E}\{\Psi_T\}.$$

By summing over all S and T we get,

$$\mathrm{Var}\{\Delta(R)\} = \sum_{S,T} \left(\mathbb{E}\{\Psi_S \Psi_T\} - \mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\}\right).$$

Specifically,

$$\begin{aligned}\mathrm{Var}\{\Delta(R)\} = &\sum_{\substack{S,T\\ S=T}} \left(\mathbb{E}\{\Psi_S \Psi_T\} - \mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\}\right) + \sum_{\substack{S,T\\ |S\cap T|=2}} \left(\mathbb{E}\{\Psi_S \Psi_T\} - \mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\}\right)\\ &+ \sum_{\substack{S,T\\ |S\cap T|=1}} \left(\mathbb{E}\{\Psi_S \Psi_T\} - \mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\}\right) + \sum_{\substack{S,T\\ |S\cap T|=0}} \left(\mathbb{E}\{\Psi_S \Psi_T\} - \mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\}\right).\end{aligned}$$

In the above equation, $|S\cap T| = 2$ represents S and T overlapping by two vertices. Likewise, $|S \cap T| = 1$ represents S and T overlapping by one vertex. Now we must prove

$$\frac{\mathrm{Var}\{\Delta(R)\}}{\mathbb{E}\{\Delta(R)\}^2} \to 0 \text{ as } n \to \infty$$

for both the $G_{n,p}$ and geometric model.

We will begin with the $G_{n,p}$ model. In the case of the $G_{n,p}$ model, we know by the definition of p and by the proof for Theorem 6.3 that $\mathbb{E}\{\Psi_S\} = p^3$ and $\mathbb{E}\{\Psi_T\} = p^3$. Likewise if $|S \cap T| = 0$ or if $|S \cap T| = 1$, $\mathbb{E}\{\Psi_S\Psi_T\} = (p^3)(p^3) = p^6$, since in each of these cases six edges are being formed with probability p. Therefore the variance for the $G_{n,p}$

model becomes

$$\mathrm{Var}\{\Delta(R)\} = \sum_{\substack{S,T\\S=T}} \left(\mathbb{E}\{\Psi_S\Psi_T\} - \mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\}\right) + \sum_{\substack{S,T\\|S\cap T|=2}} \left(\mathbb{E}\{\Psi_S\Psi_T\} - \mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\}\right).$$

If $S = T$, then $\mathbb{E}\{\Psi_S\Psi_T\} = \mathbb{E}\{\Psi_S\} = p^3$. Similarly if $|S \cap T| = 2$, then $\mathbb{E}\{\Psi_S\Psi_T\} = p^5$ since there are only five edges each determined with probability p. Therefore,

$$\mathrm{Var}\{\Delta(R)\} = \sum_{\substack{S,T\\S=T}} \left(p^3 - p^6\right) + \sum_{\substack{S,T\\|S\cap T|=2}} \left(p^5 - p^6\right).$$

Since there are $\binom{n}{3}$ cases where $S = T$ and $3\binom{n}{3}(n-3)$ cases where $|S \cap T| = 2$,

$$\mathrm{Var}\{\Delta(R)\} = \binom{n}{3}\left(p^3 - p^6\right) + 3\binom{n}{3}(n-3)\left(p^5 - p^6\right).$$

Therefore,

$$\frac{\mathrm{Var}\{\Delta(R)\}}{\mathbb{E}\{\Delta(R)\}^2} = \frac{\binom{n}{3}\left(p^3 - p^6\right) + 3\binom{n}{3}(n-3)\left(p^5 - p^6\right)}{\binom{n}{3}^2 p^6} = \frac{\left(p^3 - p^6\right) + (3n-9)\left(p^5 - p^6\right)}{\binom{n}{3}p^6}.$$

Since $p = \pi r^2$ and $r \le \frac{\sqrt{\frac{2\pi}{3} - \frac{\sqrt{3}}{2}}}{\pi}$, we are only concerned with the n values in this fraction. Thus,

$$\frac{\mathrm{Var}\{\Delta(R)\}}{\mathbb{E}\{\Delta(R)\}^2} \approx \frac{c}{n^2},$$

where c is a constant determined by the values of p. It follows that

$$\frac{\mathrm{Var}\{\Delta(R)\}}{\mathbb{E}\{\Delta(R)\}^2} \to 0 \text{ as } n \to \infty$$

holds for the $G_{n,p}$ model.

We now must prove

$$\frac{\mathrm{Var}\{\Delta(R)\}}{\mathbb{E}\{\Delta(R)\}^2} \to 0 \text{ as } n \to \infty$$

holds for the geometric model. Following a similar process, we begin with calculating the variance. We know

$$\begin{aligned}\mathrm{Var}\{\Delta(R)\} = &\sum_{\substack{S,T\\S=T}} \left(\mathbb{E}\{\Psi_S\Psi_T\} - \mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\}\right) + \sum_{\substack{S,T\\|S\cap T|=2}} \left(\mathbb{E}\{\Psi_S\Psi_T\} - \mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\}\right)\\ &+ \sum_{\substack{S,T\\S=T\setminus\{u,v\}}} \left(\mathbb{E}\{\Psi_S\Psi_T\} - \mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\}\right) + \sum_{\substack{S,T\\|S\cap T|=0}} \left(\mathbb{E}\{\Psi_S\Psi_T\} - \mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\}\right).\end{aligned}$$

In the proof for Theorem 6.4, we showed that the lower bound for the probability of a triangle in the geometric model is $\pi r^4\gamma$. Therefore, similarly to the $G_{n,p}$ model, in the $|S\cap T| = 0$ and $|S\cap T| = 1$ cases

$$(\pi r^4\gamma)(\pi r^4\gamma) = \pi^2 r^8\gamma^2 \leq \mathbb{E}\{\Psi_S\Psi_T\} = \mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\} \leq (\pi r^2)^4 = \pi^4 r^8.$$

Therefore for the geometric model, the $|S\cap T| = 0$ and $|S\cap T| = 1$ cases equal zero. Since we are dealing with estimates, we must find the upper-bound in the geometric model for

$$\frac{\mathrm{Var}\{\Delta(R)\}}{\mathbb{E}\{\Delta(R)\}^2}.$$

Therefore, we will find the upper-bound for the numerator and the lower-bound for the denominator. We determined

$$\mathrm{Var}\{\Delta(R)\} = \sum_{\substack{S,T\\S=T}} \left(\mathbb{E}\{\Psi_S\Psi_T\} - \mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\}\right) + \sum_{\substack{S,T\\|S\cap T|=2}} \left(\mathbb{E}\{\Psi_S\Psi_T\} - \mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\}\right).$$

If $S = T$, then $\pi r^4\gamma \leq \mathbb{E}\{\Psi_S\Psi_T\} = \mathbb{E}\{\Psi_S\} \leq \pi^2 r^4$. Similarly if $|S\cap T| = 2$, $\mathbb{E}\{\Psi_S\Psi_T\}$ is at its maximum when, like the upper-bound in Theorem 6.4, two points fall on top of each other, except now two points must fall in that area instead of one. Therefore in this case, $\mathbb{E}\{\Psi_S\Psi_T\} \leq (\pi r^2)(r^2\pi)^2 = \pi r^6\pi^2 = \pi^3 r^6$. Additionally, we must find the lower-bound for $\mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\}$, since it is being subtracted. This follows from Theorem 6.4, $\mathbb{E}\{\Psi_S\}\mathbb{E}\{\Psi_T\} \geq$

$(\pi r^4\gamma)(\pi r^4\gamma) = \pi^2 r^8\gamma^2$. Following the argument made for the $G_{n,p}$ model,

$$\mathrm{Var}\{\Delta(R)\} \leq \binom{n}{3}\left(\pi^2 r^4 - \pi^2 r^8\gamma^2\right) + 3\binom{n}{3}(n-3)\left(\pi^3 r^6 - \pi^2 r^8\gamma^2\right).$$

Therefore,

$$\begin{aligned}\frac{\mathrm{Var}\{\Delta(R)\}}{\mathbb{E}\{\Delta(R)\}^2} &\leq \frac{\binom{n}{3}\left(\pi^2 r^4 - \pi^2 r^8\gamma^2\right) + 3\binom{n}{3}(n-3)\left(\pi^3 r^6 - \pi^2 r^8\gamma^2\right)}{\binom{n}{3}^2\pi^2 r^8\gamma^2} \\ &= \frac{\left(\pi^2 r^4 - \pi^2 r^8\gamma^2\right) + (3n-9)\left(\pi^3 r^6 - \pi^2 r^8\gamma^2\right)}{\binom{n}{3}\pi^2 r^8\gamma^2}.\end{aligned}$$

Since π and γ are constants and $r \leq \frac{\sqrt{\frac{2\pi}{3} - \frac{\sqrt{3}}{2}}}{\pi}$, we can approximate this fraction the same way we did for $G_{n,p}$. Thus,

$$\frac{\mathrm{Var}\{\Delta(R)\}}{\mathbb{E}\{\Delta(R)\}^2} \approx \frac{c}{n^2},$$

where c is a constant determined by r. It follows that

$$\frac{\mathrm{Var}\{\Delta(R)\}}{\mathbb{E}\{\Delta(R)\}^2} \to 0 \text{ as } n \to \infty$$

holds for the geometric model. Thus, the Geometric model will asymptotically almost surely produce more triangles than the $G_{n,p}$ model while,

$$r \leq \sqrt{\frac{\frac{2\pi}{3} - \frac{\sqrt{3}}{2}}{\pi^2}} = \frac{\sqrt{\frac{2\pi}{3} - \frac{\sqrt{3}}{2}}}{\pi} \approx 0.352789.$$

Note: Although we proved this theorem for a fixed r, following the same argument we can prove it for $p = \pi r^2 = n^{-1+\varepsilon}$, where $\varepsilon > 0$. This is important as it shows that almost always there is a difference in the triangle counts of the geometric and $G_{n,p}$ models, validating us using it as a measure to determine the contributions of each model. □

6.2 Basis for Measures

It is difficult to determine whether a random graph accurately represents a fixed network because variance is inherit in random graph models even between random graphs generated from the same set of input parameters. Consequently, in order to test how successful our algorithm is in producing a random graph that accurately represents a fixed network, we will compare multiple random graphs generated from the same input parameters using $Deg_{\%}$ and $Trig_{\%}$.

As stated in section 3.1 and 3.2, our semi-norm for determining the closeness of the degree distributions is

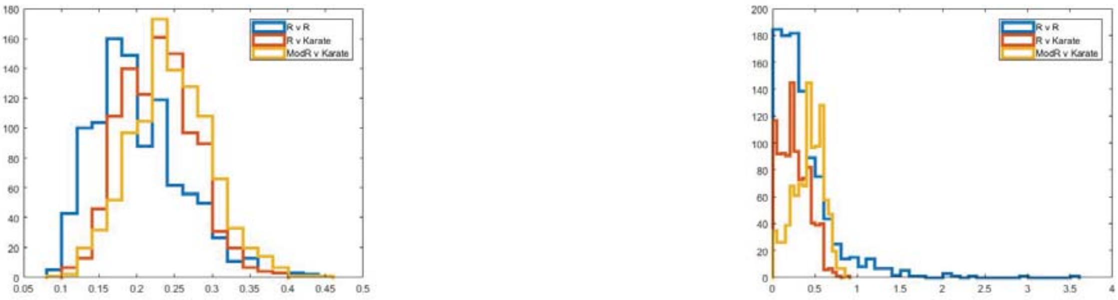
$$Deg_{\%} = \frac{\sqrt{\sum_j^n (\hat{d}_j(G) - \hat{d}_j(R))^2}}{n}.$$

and our formula for determining relative error in the triangle counts is

$$Trig_{\%} = \frac{|\Delta(R) - \Delta(G)|}{\Delta(G)}.$$

In Section 5, we tested our algorithm over several different graphs and achieved success by the inputted measures. However, it can be argued that our values for $Deg_{\%}$ and $Trig_{\%}$ were set high enough to guarantee success. This is a valid argument as a relative error of 0.50 is very poor. However despite this, we believe our method still has weight. The reason for this confidence lies in the fact that within random graphs, especially random graphs with small n values (most of the ones we tested), the variance for triangle counts is high. That is, if we were to compare two random graphs generated from the same input parameters over several trials, the $Trig_{\%}$ values would be high.

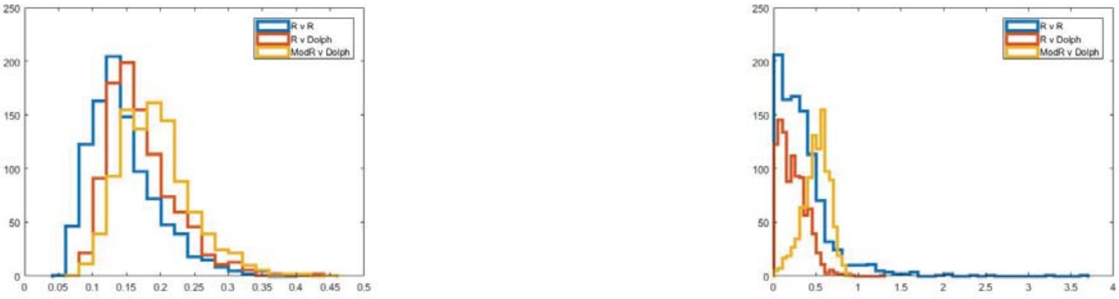
In addition to the high variance in triangle counts, it can be very difficult to get significant results on both the $Deg_{\%}$ measure and $Trig_{\%}$ measure. The $G_{\mathbf{w}}$ works very well at modeling particular degree distributions because the distribution vector can be used as $\mathbf{w}$. Additionally, the geometric model can reproduce triangle counts because its edges are based off proximity. It is very difficult to achieve any success in both of these categories. The series of figures below helps demonstrate this point. The "R v R" curve represents the comparison of two random graphs with the same input parameters over 1000 trials using both $Deg_{\%}$ and $Trig_{\%}$. The k, r, and α values used were from the table in Section 5 for each associated fixed network. The "R v G" curve represents the random graph compared to the fixed network using the two measures over 1000 trials. Finally, The "$ModR$ v G" represents the comparison of the random graph with a 10% decrease in both r and α and the fixed graph over 1000 trials. All of these comparisons were plotted using a histogram with either $Deg_{\%}$ or $Trig_{\%}$ values on the x-axis and counts on the y-axis.



(a) $Deg_{\%}$ Comparison

(b) $Trig_{\%}$ Comparison

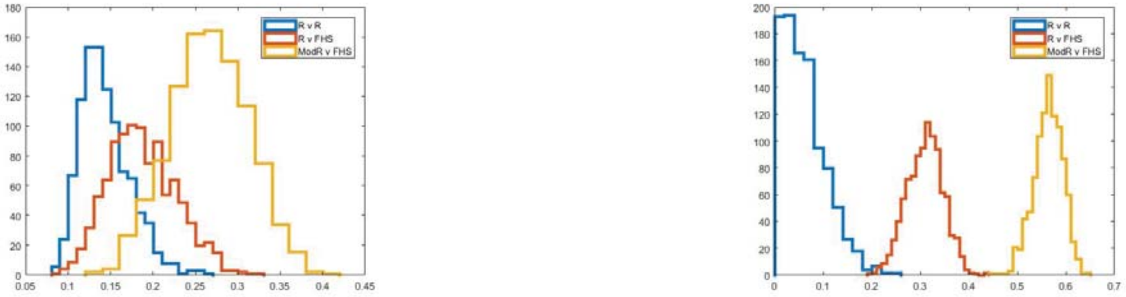
Figure 11: Zachary Karate Club Network



(a) $Deg_{\%}$ Comparison

(b) $Trig_{\%}$ Comparison

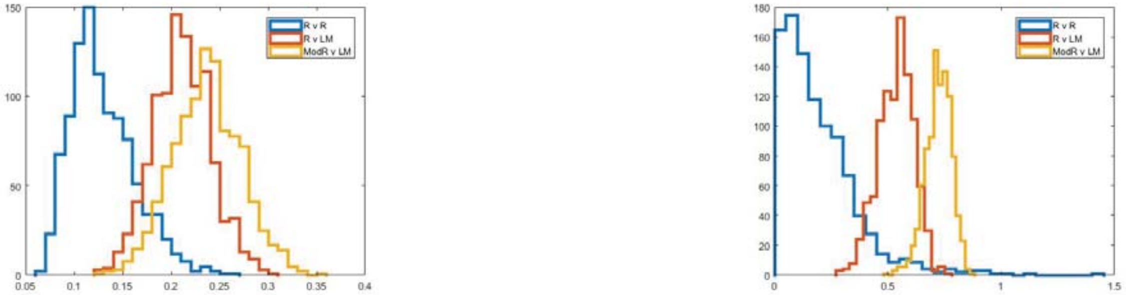
Figure 12: Dolphin Network



(a) $Deg_{\%}$ Comparison

(b) $Trig_{\%}$ Comparison

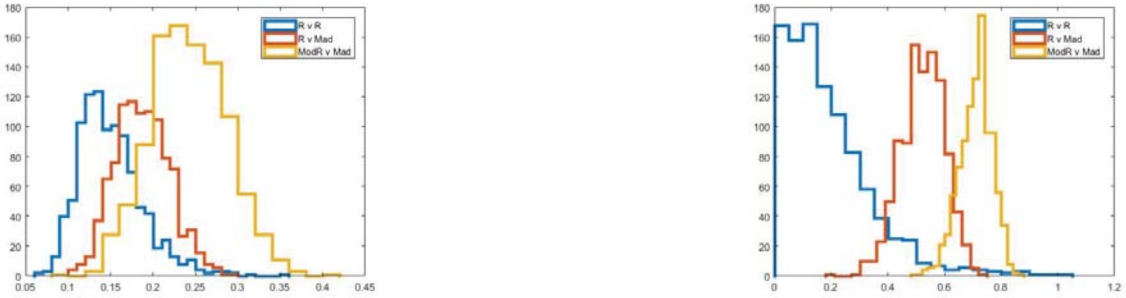
Figure 13: France High School Network



(a) $Deg_{\%}$ Comparison

(b) $Trig_{\%}$ Comparison

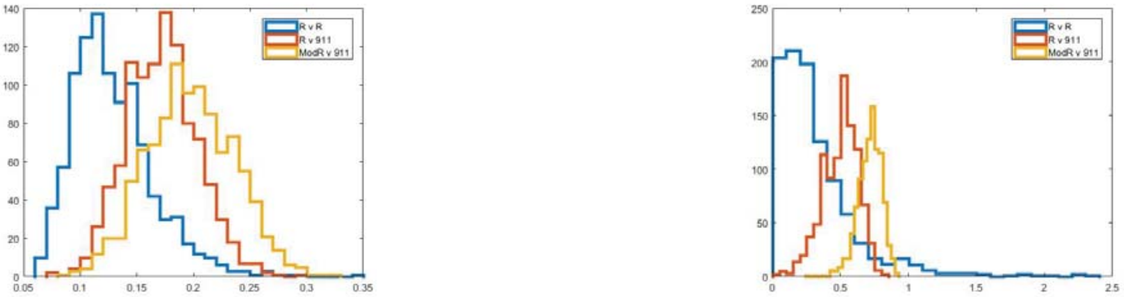
Figure 14: Les Miserables Network



(a) $Deg_{\%}$ Comparison

(b) $Trig_{\%}$ Comparison

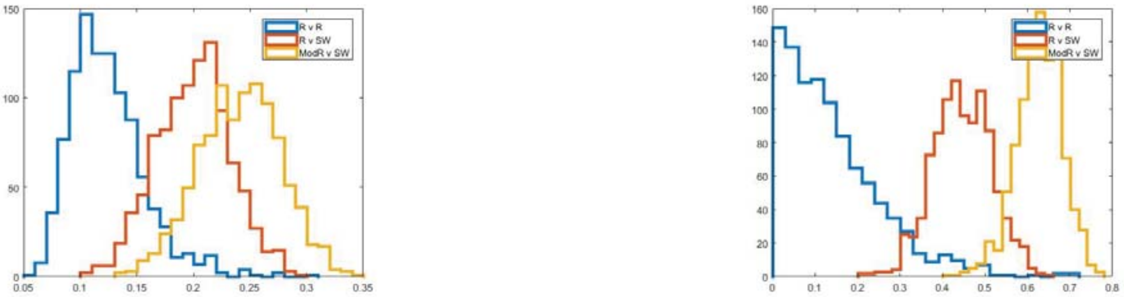
Figure 15: Madrid Train Network



(a) $Deg_{\%}$ Comparison

(b) $Trig_{\%}$ Comparison

Figure 16: 9/11 Terrorist Network



(a) $Deg_{\%}$ Comparison

(b) $Trig_{\%}$ Comparison

Figure 17: Star Wars Network

The high variance in both $Deg_{\%}$ and $Trig_{\%}$ is demonstrated well in the figures above. Obviously, our model does not work perfectly. However, although it might not seem significant, the fact that the majority of the "R v G" values fall someplace on the "R v R" graph is difficult to achieve, even if it is the far right side of the graph. Though it is not probable, this means that it falls within the realm of possibility that the fixed network was generated by the random graph. The "$ModR$ v G" curve helps demonstrate the significance of this feat as even a slight change in the input parameters can shift the curve farther to the right.

7 Conclusion and Future Directions

Broido and Clauset showed us that in actuality very few real-world networks are based solely on exponential models. However, our method of using a combination of random graph models to represent real-world networks shows promise. By broadening our scope from one type of model to three, we were able to show consistent results using a relatively simple process. Despite this fact, there are a couple of drawbacks to our process.

First, though our model has been more successful than previous attempts, it could do a lot better at modeling triangle counts in real-world networks. Additionally, our algorithm has difficulty creating distinct hubs in graphs. For example, Zachary's Karate Club network appears to have two hubs, but despite relatively consistent histogram results, we were not able to achieve as distinct of hubs in our models.

So where do we go from here? The never ending problem with attempting to model a fixed network is that it will never be perfect. However, there are a couple of things that we are working on to make it better. Currently, the algorithm only tests $k \in \{0, 1, 2\}$. The reason behind this choice was simply one of speed and efficiency. However, we believe that some of these models, like the Hamburg Terrorist network, can be modeled more accurately with higher k values, so we are currently working to allow for higher k values without decreasing efficiency. In addition to increasing the k values tested, we would like to work on a way to more accurately create hubs in the random networks. This is an issue that we have invested time into with limited results. The issue lies in the fact that in order to create a specific number of hubs in our random networks, we have to change the random graphs we are currently using, meaning at least one of the three models, $G_{\mathbf{w}}$, geometric, or preferential attachment, would have to be modified. We want to avoid altering the model too much though because then our methods for measuring the success of the models change. In order to achieve the most success with the least modification, we are currently working on modifying the preferential attachment model. Because preferential attachment graphs often have one or two vertices with high degree, we believe it is the most promising model to produce hubs. Currently with our random combination model, if a fixed network has n vertices and h number of hubs, the preferential attachment portion of that graph is generated over n vertices with a $k \in \{0, 1, 2\}$. We believe that our random combination model would be more likely to produce h hubs if the preferential attachment portion of the model had h base preferential attachment graphs each with approximately n/h vertices. This would allow for a vertex of high degree in each of the h sub-graphs. However, in order for this modification of the preferential attachment portion of the combination to factor into the algorithm, we have to increase the k values. Otherwise, the preferential attachment portion will be overpowered by the other graph types in the random combination model. However as stated earlier, perfection is impossible and every added parameter to the model brings an increase in complexity. Because of this, we believe that our current algorithm provides a good balance between simplicity and accuracy.

References

[1] N. Alon and J. H. Spencer. *The Probabilistic Method: Third Edition*. Wiley, 2008.

[2] S. Atran. John jay & artis transnational terrorism database. Technical report, Technical Report, John Jay College of Criminal Justice, 2009.

[3] A.-L. Barabási and R. Albert. Emergence of scaling in random networks. *science*, 286(5439):509–512, 1999.

[4] V. Belevitch. On the statistical laws of linguistic distributions. In *Annales de la Societe Scientifique de Bruxelles*, volume 73, pages 301–326, 1959.

[5] A. D. Broido and A. Clauset. Scale-free networks are rare. *arXiv preprint arXiv:1801.03400*, 2018.

[6] F. Chung and L. Lu. *Complex Graphs and Networks*. Number 107. American Mathematical Soc., 2006.

[7] R. Diestel. *Graph Theory*. Springer-Verlag Berlin Heidelberg, 2006.

[8] J. Fournet and A. Barrat. Contact patterns among high school students. *PloS one*, 9(9):e107878, 2014.

[9] E. Gabasova. The star wars social network. *Evelina Gabasova's Blog. Data available at: https://github. com/evelinag/StarWars-social-network/tree/master/networks*, 2015.

[10] B. Hayes. Connecting the dots. *American Scientist*, 94(5):400–404, 2006.

[11] D. E. Knuth. *The Stanford GraphBase: A Platform for Combinatorial Computing*. AcM Press New York, 1993.

[12] V. E. Krebs. Mapping networks of terrorist cells. *Connections*, 24(3):43–52, 2002.

[13] D. Lusseau, K. Schneider, O. J. Boisseau, P. Haase, E. Slooten, and S. M. Dawson. The bottlenose dolphin community of doubtful sound features a large proportion of long-lasting associations. *Behavioral Ecology and Sociobiology*, 54(4):396–405, 2003.

[14] M. Penrose et al. *Random Geometric Graphs*. Number 5. Oxford University Press, 2003.

[15] E. Ravasz and A.-L. Barabási. Hierarchical organization in complex networks. *Physical Review E*, 67(2):26–112, 2003.

[16] A. Tucker. *Applied Combinatorics: Sixth Edition*. Wiley, 2012.

[17] E. W. Weisstein. Circular segment. Visited on 04/17/19.

[18] W. W. Zachary. An information flow model for conflict and fission in small groups. *Journal of Anthropological Research*, 33(4):452–473, 1977.